Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа №6

По дисциплине «Основы алгоритмизации и программирования»

На тему «Динамические структуры данных. Односвязные списки»

Выполнил(а):

Студент(ка) 1 курса 8 группы

Семёнов Даниил Вячеславович

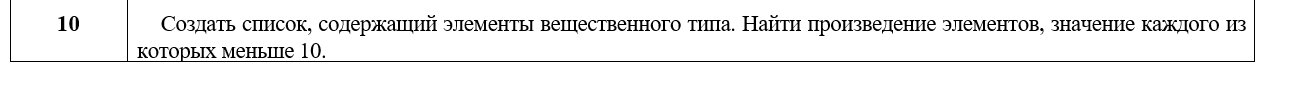
Преподаватель: асс. Андронова М.В.

Минск, 2024

4. В соответствии со своим вариантом разработать программу с использованием ***односвязного списка*** по данным, представленным в таблице ниже.

Программа должна содержать меню с пунктами: добавление элемента, удаление элемента, поиск элемента, вывод списка в консольное окно, запись списка в файл, считывание списка из файла.

Вариант 10



#include <iostream>

#include <fstream>

using namespace std;

struct list

{

float number;

list\* next;

};

void input(list\*&, float); //функция добавления элемента, передается адрес списка и символ, который добавляется

void output(list\*); //функция вывода

float del(list\*&, float); //функция удаления, передается адрес списка и символ, который удаляется

int isEmpty(list\*); //функция, которая проверяет, пуст ли список

void toFile(list\*& p); //Запись в файл

void fromFile(list\*& p); //Считывание из файла

void proizv10(list\*); //функция подсчета произведения чисел, меньших 10

void find(list\*); //функция поиска

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

float value;

do

{

cout << "1 - Ввод элементов с клавиатуры" << endl;

cout << "2 - Вывод элементов в консольное окно" << endl;

cout << "3 - Удаление заданной переменной" << endl;

cout << "4 - Поиск информации" << endl;

cout << "5 - Запись информации в файл" << endl;

cout << "6 - Чтение данных из файла" << endl;

cout << "7 - Вычисление умножения" << endl;

cout << "8 - Выход из программы" << endl;

cin >> choice;

switch (choice)

{

case 1:

int n;

cout << "Сколько чисел вы хотите ввести?" << endl;

cin >> n;

for (int i = 0;i < n;i++)

{

cout << "Введите число "; // добавить число в список

cin >> value;

input(first, value);

}

break;

case 2:

output(first);

break;

case 3:

if (!isEmpty(first)) // удалить число из списка

{

cout << "Введите удаляемое число ";

cin >> value;

if (del(first, value))

{

cout << "Удалено число " << value << endl;

output(first);

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 4:

find(first);

break;

case 5:

toFile(first);

break;

case 6:

fromFile(first);

break;

case 7:

proizv10(first); // вычисление произведения

break;

}

} while (choice!=8);

return 0;

}

void input(list\*& first, float value) //Добавление числа value в список

{

list\* newfirst = new list;

if (newfirst != NULL) //есть ли место?

{

newfirst->number = value; // присваиваем в узел значение

newfirst->next = first; // новый узел указывает на текущий первый узел

first = newfirst; // Указатель на первый элемент списка теперь указывает на новый узел

}

else

cout << "Операция добавления не выполнена" << endl;

}

float del(list\*& p, float value) // Удаление числа

{

list\* previous, \* current, \* temp;

if (value == p->number)

{

temp = p;

p = p->next; // отсоединить узел

delete temp; //освободить отсоединенный узел

return value;

}

else

{

previous = p;

current = p->next;

while (current != NULL && current->number != value)

{

previous = current;

current = current->next; // перейти к следующему

}

if (current != NULL)

{

temp = current;

previous->next = current->next;

delete(temp);

return value;

}

}

return 0;

}

int isEmpty(list\* p) //Список пустой? (1-да, 0-нет)

{

return p == NULL;

}

void output(list\* p) //Вывод списка

{

if (p == NULL)

cout << "Список пуст" << endl;

else

{

cout << "Список:" << endl;

while (p != NULL)

{

cout << "-->" << p->number;

p = p->next;

}

cout << "-->NULL" << endl;

}

}

void proizv10(list\* p) // Подсчет произведения чисел, меньших 10

{

float pr = 1;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number < 10)

pr = pr \* (p->number);

p = p->next;

}

cout << "Произведение = " << pr << endl;

}

}

void toFile(list\*& p) //Запись в файл

{

list\* temp = p;

list buf;

ofstream frm("FILE1.dat");

if (frm.fail())

{

cout << "\n Ошибка открытия файла";

exit(1);

}

while (temp)

{

buf = \*temp;

frm.write((char\*)&buf, sizeof(list));

temp = temp->next;

}

frm.close();

cout << "Список записан в файл FILE1.dat\n";

}

void fromFile(list\*& p) //Считывание из файла

{

list buf, \* first = nullptr;

ifstream frm("FILE1.dat");

if (frm.fail())

{

cout << "\n Ошибка открытия файла";

exit(1);

}

frm.read((char\*)&buf, sizeof(list));

while (!frm.eof())

{

input(first, buf.number);

cout << "-->" << buf.number;

frm.read((char\*)&buf, sizeof(list));

}

cout << "-->NULL" << endl;

frm.close();

p = first;

cout << "\nСписок считан из файла FILE1.dat\n";

}

void find(list\* p) //Функция поиска

{

float value;

cout << "Введите число для поиска ";

cin >> value;

while (p != NULL)

{

if (p->number == value)

{

cout << "Число " << value << " найдено в списке" << endl;

return;

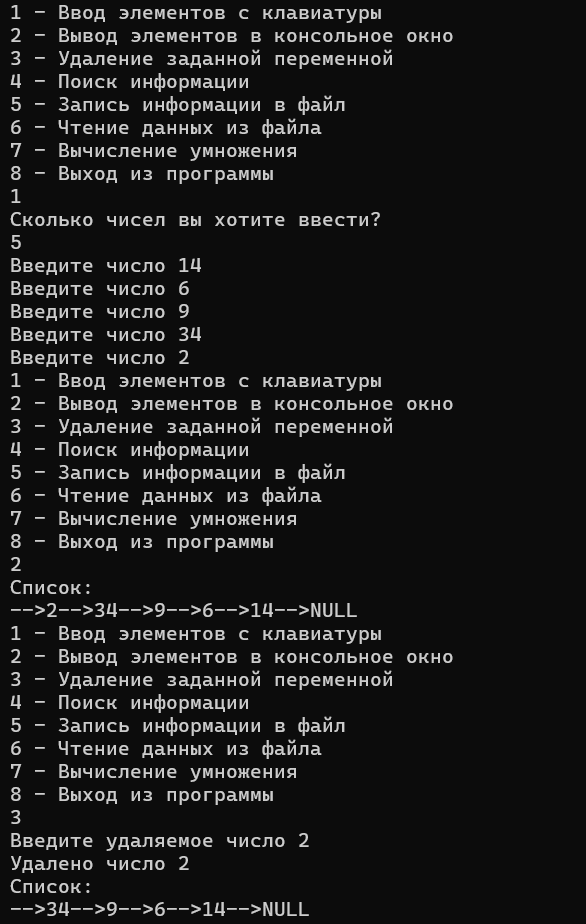
}

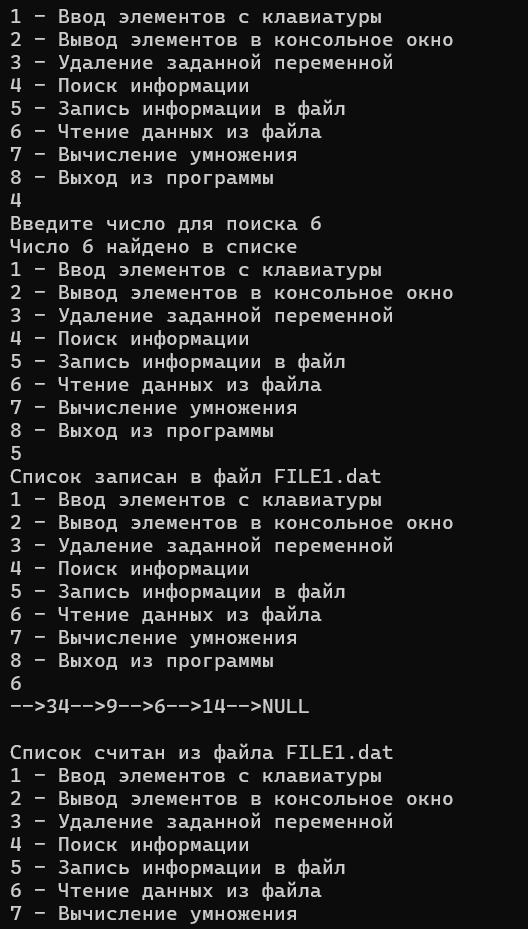
p = p->next;

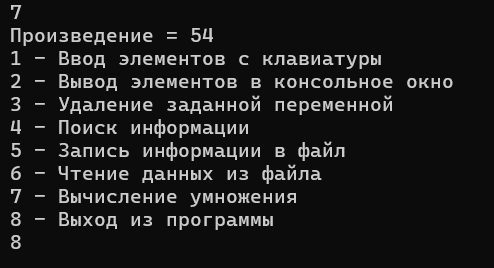
}

cout << "Число не найдено в списке" << endl;

}







Доп. Задачи

Вариант 1



#include <iostream>

#include <fstream>

using namespace std;

struct list

{

int number;

list\* next;

};

void input(list\*&, int); //функция добавления элемента, передается адрес списка и символ, который добавляется

void output(list\*); //функция вывода

float del(list\*&, int); //функция удаления, передается адрес списка и символ, который удаляется

int isEmpty(list\*); //функция, которая проверяет, пуст ли список

void toFile(list\*& p); //Запись в файл

void fromFile(list\*& p); //Считывание из файла

void sumpos(list\*); //функция подсчета суммы положительных чисел

void find(list\*); //функция поиска

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

int value;

do

{

cout << "1 - Ввод элементов с клавиатуры" << endl;

cout << "2 - Вывод элементов в консольное окно" << endl;

cout << "3 - Удаление заданной переменной" << endl;

cout << "4 - Поиск информации" << endl;

cout << "5 - Запись информации в файл" << endl;

cout << "6 - Чтение данных из файла" << endl;

cout << "7 - Вычисление суммы" << endl;

cout << "8 - Выход из программы" << endl;

cin >> choice;

switch (choice)

{

case 1:

int n;

cout << "Сколько чисел вы хотите ввести?" << endl;

cin >> n;

for (int i = 0;i < n;i++)

{

cout << "Введите число "; // добавить число в список

cin >> value;

input(first, value);

}

break;

case 2:

output(first);

break;

case 3:

if (!isEmpty(first)) // удалить число из списка

{

cout << "Введите удаляемое число ";

cin >> value;

if (del(first, value))

{

cout << "Удалено число " << value << endl;

output(first);

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 4:

find(first);

break;

case 5:

toFile(first);

break;

case 6:

fromFile(first);

break;

case 7:

sumpos(first); // вычисление произведения

break;

}

} while (choice != 8);

return 0;

}

void input(list\*& first, int value) //Добавление числа value в список

{

list\* newfirst = new list;

if (newfirst != NULL) //есть ли место?

{

newfirst->number = value; // присваиваем в узел значение

newfirst->next = first; // новый узел указывает на текующий первый узел

first = newfirst; // Указатель на первый элемент списка теперь указывает на новый узел

}

else

cout << "Операция добавления не выполнена" << endl;

}

float del(list\*& p, int value) // Удаление числа

{

list\* previous, \* current, \* temp;

if (value == p->number)

{

temp = p;

p = p->next; // отсоединить узел

delete temp; //освободить отсоединенный узел

return value;

}

else

{

previous = p;

current = p->next;

while (current != NULL && current->number != value)

{

previous = current;

current = current->next; // перейти к следующему

}

if (current != NULL)

{

temp = current;

previous->next = current->next;

delete(temp);

return value;

}

}

return 0;

}

int isEmpty(list\* p) //Список пустой? (1-да, 0-нет)

{

return p == NULL;

}

void output(list\* p) //Вывод списка

{

if (p == NULL)

cout << "Список пуст" << endl;

else

{

cout << "Список:" << endl;

while (p != NULL)

{

cout << "-->" << p->number;

p = p->next;

}

cout << "-->NULL" << endl;

}

}

void sumpos(list\* p) // Подсчет суммы положительных чисел

{

int sum = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number > 0)

sum = sum + (p->number);

p = p->next;

}

if (sum > 0)

cout << "Сумма = " << sum << endl;

else

cout <<"Положительных чисел нет"<< endl;

}

}

void toFile(list\*& p)

{

list\* temp = p;

list buf;

ofstream frm("FILE1.dat");

if (frm.fail())

{

cout << "\n Ошибка открытия файла";

exit(1);

}

while (temp)

{

buf = \*temp;

frm.write((char\*)&buf, sizeof(list));

temp = temp->next;

}

frm.close();

cout << "Список записан в файл FILE1.dat\n";

}

void fromFile(list\*& p) //Считывание из файла

{

list buf, \* first = nullptr;

ifstream frm("FILE1.dat");

if (frm.fail())

{

cout << "\n Ошибка открытия файла";

exit(1);

}

frm.read((char\*)&buf, sizeof(list));

while (!frm.eof())

{

input(first, buf.number);

cout << "-->" << buf.number;

frm.read((char\*)&buf, sizeof(list));

}

cout << "-->NULL" << endl;

frm.close();

p = first;

cout << "\nСписок считан из файла FILE1.dat\n";

}

void find(list\* p)

{

int value;

cout << "Введите число для поиска ";

cin >> value;

while (p != NULL)

{

if (p->number == value)

{

cout << "Число " << value << " найдено в списке" << endl;

return;

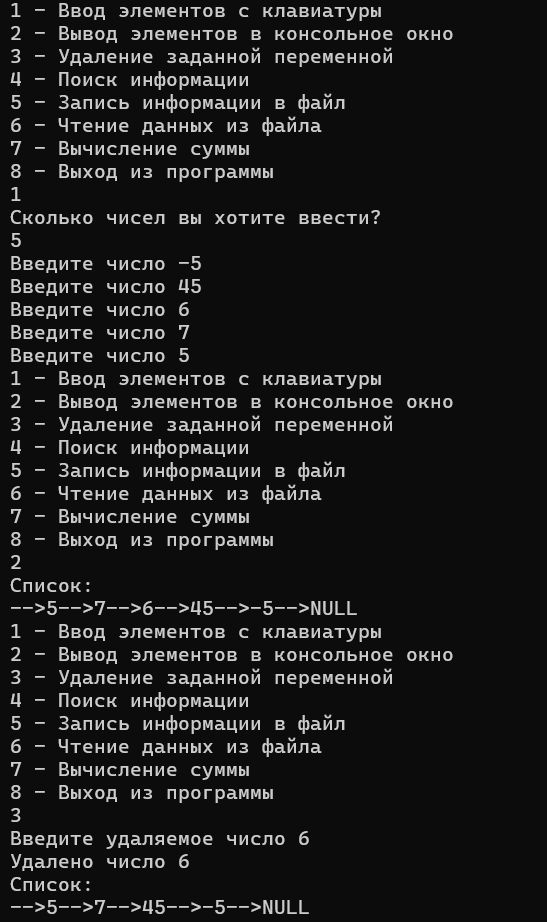
}

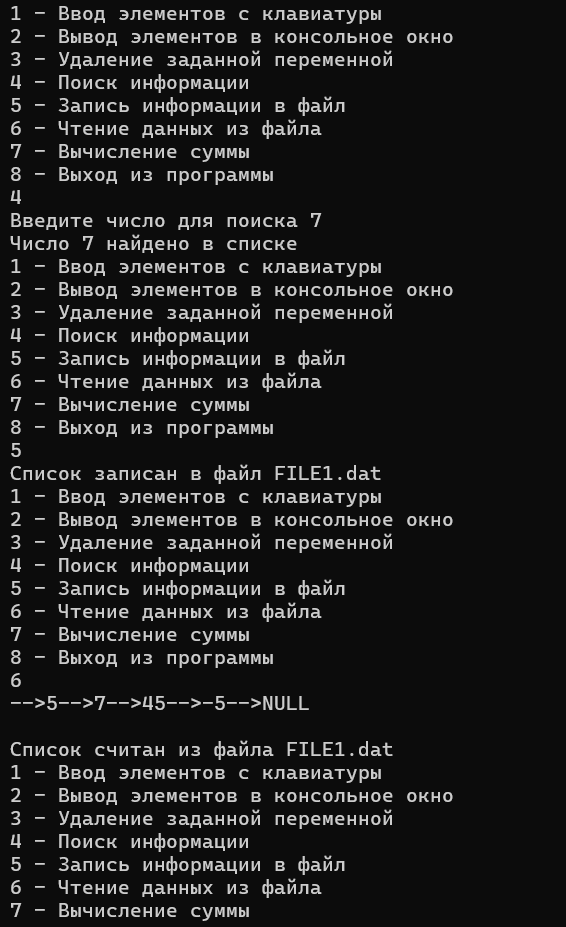
p = p->next;

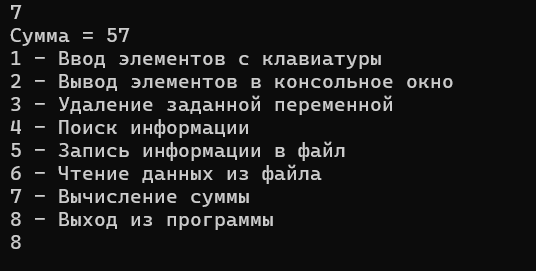
}

cout << "Число не найдено в списке" << endl;

}

****

****

****

Вариант 3



#include <iostream>

#include <fstream>

using namespace std;

struct list

{

int number;

list\* next;

};

void input(list\*&, int); //функция добавления элемента, передается адрес списка и символ, который добавляется

void output(list\*); //функция вывода

float del(list\*&, int); //функция удаления, передается адрес списка и символ, который удаляется

int isEmpty(list\*); //функция, которая проверяет, пуст ли список

void toFile(list\*& p); //Запись в файл

void fromFile(list\*& p); //Считывание из файла

void sumpos5(list\*); //функция подсчета суммы положительных чисел кратных 5

void find(list\*); //функция поиска

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

int value;

do

{

cout << "1 - Ввод элементов с клавиатуры" << endl;

cout << "2 - Вывод элементов в консольное окно" << endl;

cout << "3 - Удаление заданной переменной" << endl;

cout << "4 - Поиск информации" << endl;

cout << "5 - Запись информации в файл" << endl;

cout << "6 - Чтение данных из файла" << endl;

cout << "7 - Вычисление суммы" << endl;

cout << "8 - Выход из программы" << endl;

cin >> choice;

switch (choice)

{

case 1:

int n;

cout << "Сколько чисел вы хотите ввести?" << endl;

cin >> n;

for (int i = 0;i < n;i++)

{

cout << "Введите число "; // добавить число в список

cin >> value;

input(first, value);

}

break;

case 2:

output(first);

break;

case 3:

if (!isEmpty(first)) // удалить число из списка

{

cout << "Введите удаляемое число ";

cin >> value;

if (del(first, value))

{

cout << "Удалено число " << value << endl;

output(first);

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 4:

find(first);

break;

case 5:

toFile(first);

break;

case 6:

fromFile(first);

break;

case 7:

sumpos5(first); // вычисление произведения

break;

}

} while (choice != 8);

return 0;

}

void input(list\*& first, int value) //Добавление числа value в список

{

list\* newfirst = new list;

if (newfirst != NULL) //есть ли место?

{

newfirst->number = value; // присваиваем в узел значение

newfirst->next = first; // новый узел указывает на текующий первый узел

first = newfirst; // Указатель на первый элемент списка теперь указывает на новый узел

}

else

cout << "Операция добавления не выполнена" << endl;

}

float del(list\*& p, int value) // Удаление числа

{

list\* previous, \* current, \* temp;

if (value == p->number)

{

temp = p;

p = p->next; // отсоединить узел

delete temp; //освободить отсоединенный узел

return value;

}

else

{

previous = p;

current = p->next;

while (current != NULL && current->number != value)

{

previous = current;

current = current->next; // перейти к следующему

}

if (current != NULL)

{

temp = current;

previous->next = current->next;

delete(temp);

return value;

}

}

return 0;

}

int isEmpty(list\* p) //Список пустой? (1-да, 0-нет)

{

return p == NULL;

}

void output(list\* p) //Вывод списка

{

if (p == NULL)

cout << "Список пуст" << endl;

else

{

cout << "Список:" << endl;

while (p != NULL)

{

cout << "-->" << p->number;

p = p->next;

}

cout << "-->NULL" << endl;

}

}

void sumpos5(list\* p) // Подсчет суммы положительных чисел кратных 5

{

int sum = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number > 0 && p->number%5==0)

sum = sum + (p->number);

p = p->next;

}

if (sum > 0)

cout << "Сумма = " << sum << endl;

else

cout << "Положительных чисел кратных 5 - нет" << endl;

}

}

void toFile(list\*& p)

{

list\* temp = p;

list buf;

ofstream frm("FILE1.dat");

if (frm.fail())

{

cout << "\n Ошибка открытия файла";

exit(1);

}

while (temp)

{

buf = \*temp;

frm.write((char\*)&buf, sizeof(list));

temp = temp->next;

}

frm.close();

cout << "Список записан в файл FILE1.dat\n";

}

void fromFile(list\*& p) //Считывание из файла

{

list buf, \* first = nullptr;

ifstream frm("FILE1.dat");

if (frm.fail())

{

cout << "\n Ошибка открытия файла";

exit(1);

}

frm.read((char\*)&buf, sizeof(list));

while (!frm.eof())

{

input(first, buf.number);

cout << "-->" << buf.number;

frm.read((char\*)&buf, sizeof(list));

}

cout << "-->NULL" << endl;

frm.close();

p = first;

cout << "\nСписок считан из файла FILE1.dat\n";

}

void find(list\* p)

{

int value;

cout << "Введите число для поиска ";

cin >> value;

while (p != NULL)

{

if (p->number == value)

{

cout << "Число " << value << " найдено в списке" << endl;

return;

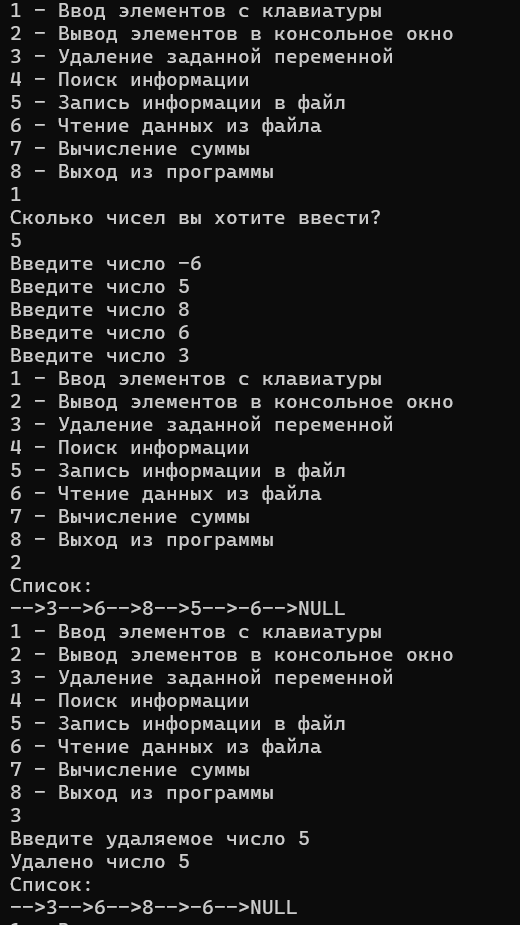
}

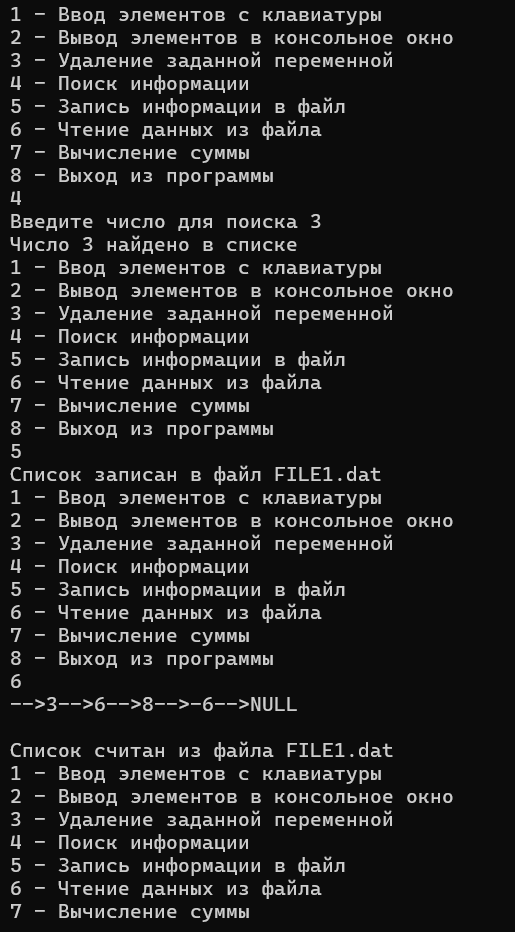
p = p->next;

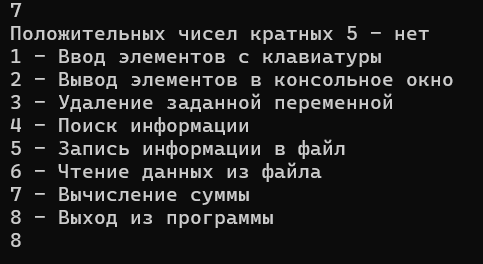
}

cout << "Число не найдено в списке" << endl;

}







Вариант 8



#include <iostream>

#include <fstream>

using namespace std;

struct list

{

int number;

list\* next;

};

void input(list\*&, int); //функция добавления элемента, передается адрес списка и символ, который добавляется

void output(list\*); //функция вывода

float del(list\*&, int); //функция удаления, передается адрес списка и символ, который удаляется

int isEmpty(list\*); //функция, которая проверяет, пуст ли список

void toFile(list\*& p); //Запись в файл

void fromFile(list\*& p); //Считывание из файла

void sumpos2(list\*); //функция подсчета суммы отрицательных чисел кратных 2

void find(list\*); //функция поиска

int main()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

int value;

do

{

cout << "1 - Ввод элементов с клавиатуры" << endl;

cout << "2 - Вывод элементов в консольное окно" << endl;

cout << "3 - Удаление заданной переменной" << endl;

cout << "4 - Поиск информации" << endl;

cout << "5 - Запись информации в файл" << endl;

cout << "6 - Чтение данных из файла" << endl;

cout << "7 - Вычисление суммы" << endl;

cout << "8 - Выход из программы" << endl;

cin >> choice;

switch (choice)

{

case 1:

int n;

cout << "Сколько чисел вы хотите ввести?" << endl;

cin >> n;

for (int i = 0;i < n;i++)

{

cout << "Введите число "; // добавить число в список

cin >> value;

input(first, value);

}

break;

case 2:

output(first);

break;

case 3:

if (!isEmpty(first)) // удалить число из списка

{

cout << "Введите удаляемое число ";

cin >> value;

if (del(first, value))

{

cout << "Удалено число " << value << endl;

output(first);

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 4:

find(first);

break;

case 5:

toFile(first);

break;

case 6:

fromFile(first);

break;

case 7:

sumpos2(first); // вычисление произведения

break;

}

} while (choice != 8);

return 0;

}

void input(list\*& first, int value) //Добавление числа value в список

{

list\* newfirst = new list;

if (newfirst != NULL) //есть ли место?

{

newfirst->number = value; // присваиваем в узел значение

newfirst->next = first; // новый узел указывает на текующий первый узел

first = newfirst; // Указатель на первый элемент списка теперь указывает на новый узел

}

else

cout << "Операция добавления не выполнена" << endl;

}

float del(list\*& p, int value) // Удаление числа

{

list\* previous, \* current, \* temp;

if (value == p->number)

{

temp = p;

p = p->next; // отсоединить узел

delete temp; //освободить отсоединенный узел

return value;

}

else

{

previous = p;

current = p->next;

while (current != NULL && current->number != value)

{

previous = current;

current = current->next; // перейти к следующему

}

if (current != NULL)

{

temp = current;

previous->next = current->next;

delete(temp);

return value;

}

}

return 0;

}

int isEmpty(list\* p) //Список пустой? (1-да, 0-нет)

{

return p == NULL;

}

void output(list\* p) //Вывод списка

{

if (p == NULL)

cout << "Список пуст" << endl;

else

{

cout << "Список:" << endl;

while (p != NULL)

{

cout << "-->" << p->number;

p = p->next;

}

cout << "-->NULL" << endl;

}

}

void sumpos2(list\* p) // Подсчет суммы отрицательных чисел кратных 2

{

int sum = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number < 0 && p->number % 2 == 0)

sum = sum + (p->number);

p = p->next;

}

if (sum < 0)

cout << "Сумма = " << sum << endl;

else

cout << "Отрицательных чисел кратных 2 - нет" << endl;

}

}

void toFile(list\*& p)

{

list\* temp = p;

list buf;

ofstream frm("FILE1.dat");

if (frm.fail())

{

cout << "\n Ошибка открытия файла";

exit(1);

}

while (temp)

{

buf = \*temp;

frm.write((char\*)&buf, sizeof(list));

temp = temp->next;

}

frm.close();

cout << "Список записан в файл FILE1.dat\n";

}

void fromFile(list\*& p) //Считывание из файла

{

list buf, \* first = nullptr;

ifstream frm("FILE1.dat");

if (frm.fail())

{

cout << "\n Ошибка открытия файла";

exit(1);

}

frm.read((char\*)&buf, sizeof(list));

while (!frm.eof())

{

input(first, buf.number);

cout << "-->" << buf.number;

frm.read((char\*)&buf, sizeof(list));

}

cout << "-->NULL" << endl;

frm.close();

p = first;

cout << "\nСписок считан из файла FILE1.dat\n";

}

void find(list\* p)

{

int value;

cout << "Введите число для поиска ";

cin >> value;

while (p != NULL)

{

if (p->number == value)

{

cout << "Число " << value << " найдено в списке" << endl;

return;

}

p = p->next;

}

cout << "Число не найдено в списке" << endl;

}

